Evolutionary Synthesis of Cube Root Computational Circuit Using Graph Hybrid Estimation of Distribution Algorithm

Josef SLEZÁK, Jiří PETRŽELA

Dept. of Radio Electronics, Brno University of Technology, Technická 12, 616 00 Brno, Czech Republic

xsleza08@stud.feec.vutbr.cz, petrzelj@feec.vutbr.cz

Abstract. The paper is focused on evolutionary synthesis of analog circuit realization of cube root function using proposed Graph Hybrid Estimation of Distribution Algorithm. The problem of cube root function circuit realization was adopted to demonstrate synthesis capability of the proposed method. Individuals of the population of the proposed method which represent promising topologies are encoded using graphs and hypergraphs. Hybridization with local search algorithm was used. The proposed method employs univariate probabilistic model.

Keywords
Automated analog circuit synthesis, evolutionary algorithm, analog circuit design, estimation of distribution algorithm, computational circuit, univariate marginal distribution algorithm.

1. Introduction
Design of analog circuits is traditionally a domain of experienced designers and usually is viewed as a kind of art where designer’s intuition involved in the design process is very important factor. Since design of analog circuits is an expensive and time consuming process there is effort to automatize the process using automated computer analog circuit design tools.

There have been published number of papers focusing on the subject of automated analog circuit design employing variety of optimization methods.

In [4] Koza et al. presented method of automated passive analog circuit synthesis system employing genetic programming where analog electronic circuits were represented as tree structures.

Passive circuits synthesis method employing hybrid genetic algorithm combined with local search algorithm and direct encoding method was published by Grimbleby in [5]. The synthesis was performed in two steps. In the first one the topology was selected and its simulatability was verified using symbolic calculation routine. In the second step the parameters (values of the components) were determined using numerical optimization method.

Method of passive analog circuits synthesis based on genetic algorithm with developmental encoding was presented by Lohn and Colombano in [6]. The basic principle of the developmental encoding is to use sequence of circuit-building instructions (OP codes) which construct the topology of the circuit. The motivation for using development encoding method was demand to decrease number of dead (nonsimulatable) individuals created after recombination phase of classic genetic algorithm. On the other hand the developmental encoding method can restrict possible encodable analog circuit topologies in some cases.

More advanced approach of synthesis of passive and also active analog circuits was proposed by Zebulum et al. who employed genetic algorithm with variable chromosome representation [7]. Besides the main chromosome vector containing the analog circuit structure information the genetic algorithm utilizes also mask vector which is used to define coding and noncoding segments of the main chromosome. There were proposed three approaches called ILG, OLG and UDIP which were used for manipulation of the bits of the mask vector. The method was also used for unconstrained evolution of analog computational QR circuit [2].

Mattiussi has proposed method called analog genetic encoding (AGE) which is able to synthesize active analog circuits and neural networks [8]. The system employs encoding method based on the principles of biological chromosomes.

Das and Vemuri have proposed several methods of automated analog circuit synthesis. The first method called GAPSYS was able to synthesize only passive analog circuits [9]. Another two methods divide the synthesis into two separate processes - selection topology and sizing of the components. In the method presented in [10] the selection of the topology is realized using adaptively generated building blocks. Evolutionary electronics synthesis method using graph grammar based approach was presented in [11].
Analog circuits encoding method based on adjacency matrix representation and special type of crossover was presented by Mesquite et al. in [12]. Compared to incidence matrix representation the proposed method is able to preserve topologies of both parental circuits and to connect them in a meaningful way through subset of nodes [12].

Analog circuits synthesis using simulated annealing method was presented in [13], [14].

Recently Estimation of Distribution Algorithms (EDA) [15] have shown their superior performance compared to classical genetic algorithms. Univariate Marginal Distribution Algorithm (UMDA) [16] which is the simplest version of EDA was employed in evolutionary electronics system presented by Zinchenko [17]. The proposed system was verified on the problem of synthesis of low pass filter. Another application of UMDA in analog circuit synthesis method was presented by Torres [18].

Presented paper is focused on synthesis of cube root computational circuit based on Estimation of Distribution Algorithm. Since the individuals of the population are represented as graphs and hypergraphs and hybridization with local search algorithm is used the proposed algorithm is called Graph Hybrid Estimation of Distribution Algorithm (GhEDA). The method employs univariate probabilistic model.

2. Definition of the Problem

The problem of the synthesis of analog circuit realization of cube root function was introduced by Koza et al. in [1]. The problem was also adopted in [2]. The target voltage response of the desired circuit is

\[ U_2 = \sqrt[3]{U_1}. \]  

(1)

In other words the goal of the synthesis is to design analog circuit in which output voltage \( U_2 \) is cube root of its input voltage \( U_1 \).

3. Introduction of Graph Estimation of Distribution Algorithm

Synthesis capability of the proposed GhEDA method will be demonstrated on the problem of circuit realization of cube root function. The cube root function circuit realization consists of bipolar transistors NPN and PNP, resistors and positive and negative voltage sources. The goal of the synthesis is to design the topology of connection of the transistors NPN and PNP, topology of connection of the resistors, parameters of the resistors (values) and to define nodes of connection of the positive and the negative voltage sources. Pseudo-code of the proposed method is presented in Fig. 1. The proposed algorithm is Estimation of Distribution Algorithm type. Therefore recombination phase as used in genetic algorithms is replaced by building and sampling of the probabilistic model. No recombination operators such as crossover and mutation are used.

**step0**: Initialize population \( P \) of \( m \) individuals.

**step1**: According to selection method select population \( P_{sel} \).

**step2**: Build probabilistic model \( M \) of selected population \( P_{opt} \).

**step3**: Using probabilistic model \( M \) generate set of new samples \( P_{samp} \) consisting of \( d \) individuals.

**step4**: Using cost objective function evaluate cost values of set of new samples \( P_{samp} \).

**step5**: Based on \( P \) and \( P_{samp} \) create new population \( P_{new} \) and replace old population \( (P := P_{new}) \).

**step6**: According to topologies of \( n_{opt} \) randomly selected individuals of \( P \) optimize parameters storage \( P_{St} \). Go to **step1**.

Fig. 1. Pseudo code of the proposed method.

Initial population \( P \) consisting of \( m \) individuals is set randomly respecting maximal number of components of every type \( n_{npp}, n_{pnp}, n_{nre}, n_{vccp} \) and \( n_{vccn} \). Parameters storage \( P_{St} \) is initialized randomly with uniform distribution. Detailed description of the encoding method and parameters storage \( P_{St} \) is presented in Section 4.

After evaluation of the cost values of population \( P \), selected population \( P_{sel} \) is formed. Tournament selection method with tournament size 2 is used.

In the learning phase probabilistic model \( M \) of selected population \( P_{sel} \) is created. Marginal frequencies of the components included in selected population \( P_{sel} \) are calculated. Every single component connected to a specific set of connection nodes is represented by corresponding edge of the graph (resistors and positive and negative voltage sources) or hyperedge of the hypergraph (transistors NPN and PNP). Therefore marginal frequencies of the components correspond to the marginal frequencies of the edges of the graphs and the hyperedges of the hypergraphs encoded in the individuals of selected population \( P_{sel} \). Detailed description of the learning phase is presented in Section 5.

In the next phase probabilistic model \( M \) is used to generate population of new samples of solutions \( P_{samp} \) which consists of \( d \) individuals. Detailed description of the sampling phase is described in Section 6.

New individuals are simulated and theirs cost values are calculated using objective function described in Section 7.

In the replacement phase new population \( P_{new} \) is formed of the best \( m - d \) individuals of current population \( P \) and whole population of new samples \( P_{samp} \). Afterwards current population \( P \) is replaced by new population \( P_{new} \) \( (P := P_{new}) \).

In the optimization phase the local search algorithm tries to improve (decrease) cost values of \( n_{opt} \) randomly selected individuals of population \( P \). Detailed description of the optimization phase is presented in Section 8.
4. Encoding Method

Graphs are the most straightforward method of representation of the topology of analog circuits. The desired circuit realization of cube root function consists of resistors, bipolar transistors NPN and PNP and positive and negative voltage sources. As will be described below the topology of connection of resistors and connection of the positive and the negative voltage sources are represented by corresponding graphs. Topologies of connection of transistors NPN and PNP are represented by 3-uniform hypergraphs.

Maximal complexity of the desired analog circuit is defined by maximal number of nodes $n_{nod}$ and maximal number of transistors NPN, transistors PNP and resistors denoted as $n_{npn}$, $n_{pnp}$, $n_{res}$. Maximal number of nodes connected to positive and negative voltage sources are denoted as $n_{vccp}$ and $n_{vccn}$ respectively. Every individual of population $P$ consists informations about topology of connection of transistors NPN and PNP, topology of connection of resistors and connection of positive and negative voltage sources. Parameters of the resistors are stored in parameters storage $PS$ which is described in Section 8.

The topology of resistors is represented by simple undirected graph $G_{res}$. Since maximal circuit complexity is restricted to $n_{nod}$ nodes graph $G_{res}$ is always subgraph of complete graph $G_{resc}$ which includes $n_{nod}$ vertices and $n_{edges} = (n_{nod} - 1)/2$ edges. Complete graph $G_{resc}$ for $n_{nod} = 4$ and corresponding topology of the resistors are presented in Fig. 2a and Fig. 2b respectively. Example of graph $G_{res}$ and corresponding topology of resistors are presented in Fig. 3a and Fig. 3b.

Graph $G_{res}$ is defined by its characteristic vector. Maximal number of the edges of graph $G_{res}$ is defined by number of edges $n_{edges}$ of corresponding complete graph $G_{resc}$. Characteristic vector of graph $G_{res}$ can be defined as binary vector $e_{res}$ of length $n_{edges}$ bits. Every single bit of $e_{res}$ corresponds to including or not including corresponding edge of complete graph $G_{resc}$ in its subgraph $G_{res}$. Characteristic vector $e_{res}$ of graph $G_{res}$ is presented in Fig. 3b.

Assignment of the edges to the vertices for graphs $G_{res}$ and $G_{resc}$ and assignment of resistors to nodes for corresponding circuits (Fig. 2b and Fig. 3b) are presented in Tab. 1.

<table>
<thead>
<tr>
<th>edge (resistor)</th>
<th>vertex 1 (node 1)</th>
<th>vertex 2 (node 2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_1 (R_1)$</td>
<td>$v_1 (n_0)$</td>
<td>$v_1 (n_1)$</td>
</tr>
<tr>
<td>$e_2 (R_2)$</td>
<td>$v_1 (n_0)$</td>
<td>$v_2 (n_2)$</td>
</tr>
<tr>
<td>$e_3 (R_3)$</td>
<td>$v_1 (n_0)$</td>
<td>$v_3 (n_3)$</td>
</tr>
<tr>
<td>$e_4 (R_4)$</td>
<td>$v_1 (n_1)$</td>
<td>$v_3 (n_3)$</td>
</tr>
<tr>
<td>$e_5 (R_5)$</td>
<td>$v_1 (n_1)$</td>
<td>$v_3 (n_3)$</td>
</tr>
<tr>
<td>$e_6 (R_6)$</td>
<td>$v_2 (n_2)$</td>
<td>$v_3 (n_3)$</td>
</tr>
</tbody>
</table>

Fig. 2. a) Graph $G_{res}$ b) analog circuit corresponding to $G_{res}$.

Fig. 3. a) graph $G_{res}$ b) analog circuit corresponding to $G_{res}$ and encoding vector of $G_{res}$.

Fig. 4. Complete 3-uniform hypergraph $G_{npnc}$ for $n_{nod} = 4$. 

Topology of transistors NPN is represented by labeled 3-uniform hypergraph $G_{npn}$ and is restricted to $n_{nod}$ nodes. Example of labeled 3-uniform hypergraph and corresponding analog circuit are presented in Fig. 4, Fig. 5 and Fig. 6.
Complete 3-uniform hypergraph $G_{npnc}$ for $n_{nod} = 4$ is presented in Fig. 4. Larger white circles represent the vertices of the hypergraph. Smaller black circles represent the hyperedges. Corresponding analog circuit is presented in Fig. 5.

Assignment of the hyperedges to the vertices for hypergraphs $G_{npnc}$ and $G_{npn}$ and assignment of the pins of the transistors to the nodes for corresponding circuits (Fig. 5 and Fig. 7) are presented in Tab. 2.

Since “rotation” labels are not specified in complete 3-uniform hypergraph $G_{npnc}$, generalized three-ports admittances $Y_1$ to $Y_4$ are used in the place of the transistors in Fig. 5. Example of labeled 3-uniform hypergraph $G_{npn}$ is presented in Fig. 6. Numbers in the brackets behind the names of the hyperedges define the labels of the hyperedges. For hyperedges $e_1$ and $e_3$ of hypergraph $G_{npn}$, labels “rotation” are set to 1 and 3 respectively. Assignment of the labels of the hyperedges to “rotation” of the transistors is defined in Tab. 3.

Since every possible configuration of hypergraph $G_{npn}$ is subgraph of complete 3-uniform hypergraph $G_{npnc}$, characteristic vector of hypergraph $G_{npn}$ can be defined as binary vector $e_{npn}$ of length $n_{edgnpc} = n_{nod}^3 - 3n_{nod}^2 + 3n_{nod} - n_{nod}$ bits. Every single bit of $e_{npn}$ corresponds to including or not including corresponding hyperedge of complete hypergraph $G_{npnc}$ in its subhypergraph $G_{npn}$. Encoding vector $e_{npn}$ is further extended to include information about “rotation” of the encoded transistors. There are six possible combinations of connection of the transistor to three nodes. Therefore the final encoding vector $e_{npn}$ is defined as binary vector of length $6n_{edgnpc}$. Encoding vector $e_{npn}$ of hypergraph $G_{npn}$ presented in Fig. 6 is presented in Fig. 8.

The topology of PNP transistors is represented by labeled 3-uniform hypergraph $G_{pnp}$ and encoded by vector

---

**Tab. 2.** Assignment of the hyperedges to the vertices for hypergraphs $G_{npnc}$ and $G_{npn}$ and assignment of the pins of the transistors to the nodes for circuits in Fig. 5 and Fig. 7.

<table>
<thead>
<tr>
<th>hyperedge</th>
<th>vertex 1</th>
<th>vertex 2</th>
<th>vertex 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e_1$</td>
<td>$v_1 (n_1)$</td>
<td>$v_2 (n_2)$</td>
<td>$v_3 (n_3)$</td>
</tr>
<tr>
<td>$e_2$</td>
<td>$v_1 (n_1)$</td>
<td>$v_2 (n_2)$</td>
<td>$v_4 (n_4)$</td>
</tr>
<tr>
<td>$e_3$</td>
<td>$v_2 (n_2)$</td>
<td>$v_3 (n_3)$</td>
<td>$v_4 (n_4)$</td>
</tr>
<tr>
<td>$e_4$</td>
<td>$v_1 (n_1)$</td>
<td>$v_3 (n_3)$</td>
<td>$v_4 (n_4)$</td>
</tr>
</tbody>
</table>

**Tab. 3.** Assignment of the labels of the hyperedges to the corresponding connection nodes of the transistors.

<table>
<thead>
<tr>
<th>label (“rotation”)</th>
<th>node 1</th>
<th>node 2</th>
<th>node 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>B</td>
<td>C</td>
<td>E</td>
</tr>
<tr>
<td>2</td>
<td>B</td>
<td>E</td>
<td>B</td>
</tr>
<tr>
<td>3</td>
<td>E</td>
<td>E</td>
<td>C</td>
</tr>
<tr>
<td>4</td>
<td>E</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>5</td>
<td>E</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>6</td>
<td>E</td>
<td>B</td>
<td>C</td>
</tr>
</tbody>
</table>

**Fig. 8.** Encoding vector $e_{npn}$ of hypergraph $G_{npn}$.
The parameters of the resistors (the values of the resistors) are stored in parameters storage PS which is vector of real numbers of length n_{edges}. Vector PS includes value for every possible resistor connected to nodes n_1 and n_2, where n_1 \in \{0, 1, \ldots, n_{mod} - 1\} and n_2 \in \{0, 1, \ldots, n_{mod} - 1\}.

During every single evaluation of the objective function the cost value is obtained based on two types of information. The first one informs about the topology and is stored in encoding vectors of the individuals (\(e_{res}, e_{pnp}, e_{pmp}, e_{vccp}, e_{vccn}\)) in population \(P\). The second one informs about the parameters of the encoded resistors and is stored in parameters storage \(PS\).

The only way how to modify the values of parameters storage \(PS\) is execution of the local search algorithm (LSA) in the optimization phase (step 6 in Fig. 1). Synthesis process consists of mutual interaction between selection of the promising topologies (step 1 in Fig. 1) and optimization of the values of parameters storage \(PS\). In the optimization phase LSA tries to optimize the values of \(PS\) to adapt them to the promising topologies selected in the selection phase. This way the values stored in parameters storage \(PS\) are evolved during the whole synthesis process.

5. Learning of the Probabilistic Model

For every single component type (transistors NPN, transistors PNP, resistors, positive voltage sources, negative voltage sources) marginal frequencies of the edges and hyperedges contained in current selected population \(P_{sel}\) are calculated and saved in vectors \(v_{pnp}, v_{pmp}, v_{res}, v_{vccp}, v_{vccn}\) which are encoded the same way as encoding vectors \(e_{pnp}, e_{pmp}, e_{res}, e_{vccp}, e_{vccn}\). The values of vectors \(v_{pnp}, v_{pmp}, v_{res}, v_{vccp}, v_{vccn}\) represent numbers of appearing of the corresponding edges in current selected population \(P_{sel}\). Examples of vectors \(v_{pnp}, v_{pmp}, v_{res}, v_{vccp}, v_{vccn}\) are presented in Fig. 11.

\[
\begin{array}{cccc}
Y_1 & Y_2 & Y_3 & Y_4 \\
1 & 2 & 3 & 4 \\
3 & 4 & 5 & 6 \\
6 & 1 & 2 & 3 \\
5 & 6 & 1 & 2 \\
4 & 5 & 6 & 1 \\
\end{array}
\]

\[
\begin{array}{cccc}
Y_1 & Y_2 & Y_3 & Y_4 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\end{array}
\]

\[
\begin{array}{cccc}
Y_1 & Y_2 & Y_3 & Y_4 \\
2 & 9 & 5 & 4 \\
4 & 7 & 3 & 1 \\
\end{array}
\]

\[
\begin{array}{cccc}
Y_1 & Y_2 & Y_3 & Y_4 \\
1 & 2 & 7 & 2 \\
1 & 2 & 7 & 2 \\
\end{array}
\]

\[
\begin{array}{cccc}
Y_1 & Y_2 & Y_3 & Y_4 \\
3 & 6 & 2 & 1 \\
3 & 6 & 2 & 1 \\
\end{array}
\]

Fig. 11. Examples of vector \(v_{pnp}\) and \(v_{pmp}\) (a), \(v_{res}\) (b), \(v_{vccp}\) (c) and \(v_{vccn}\) (d).

---

\(e_{pnp}\) exactly the same way as was described above for the topology of NPN transistors.

The last type of information which has to be encoded is connection of positive and negative voltage sources what is represented by graphs \(G_{vccp}\) and \(G_{vccn}\).

As can be seen in example in Fig. 9a graph \(G_{vccp}\) includes vertex \(V_{vccp}\) which represents positive voltage source. Edges between vertices \(V_{vccp}\) and \(v_1\) and \(v_3\) represent connection of positive voltage source \(V_{vccp}\) to nodes \(n_1\) and \(n_3\).

Similarly in graph \(G_{vccn}\) vertex \(V_{vccn}\) is connected to vertices \(v_2\) and \(v_4\) what corresponds to connection negative voltage source \(V_{vccn}\) to nodes \(n_2\) and \(n_4\) (Fig. 9a). Schematic representation of analog circuit corresponding to graphs in Fig. 9a is presented in Fig. 9b.

\[V_{vccp} = [2 9 5 4 7 3] \quad V_{vccn} = [3 6 2 5 2 1]\]

![Fig. 9. a) Graphs \(G_{vccp}\) and \(G_{vccn}\) b) connection of voltage sources defined by graphs \(G_{vccp}\) and \(G_{vccn}\).](image)

Encoding vectors \(e_{vccp}\) and \(e_{vccn}\) of graphs \(G_{vccp}\) and \(G_{vccn}\) are represented by binary vectors of length \(n_{mod}\), where every single bit represents including or not including of an edge between voltage source \(V_{vccp}\) or \(V_{vccn}\) and corresponding vertex \((v_1\) to \(v_5\) in the example). Encoding vectors \(e_{vccp}\) and \(e_{vccn}\) are presented in Fig. 10.

\[
e_{vccp} = [1 0 1 0 0 0] \\
e_{vccn} = [0 1 0 1 0 0]
\]

![Fig. 10. Encoding vector \(e_{vccp}\) of graph \(G_{vccp}\) and encoding vector \(e_{vccn}\) of graph \(G_{vccn}\).](image)
For example \( v_{npn}(3) = 4 \) (number 4 in the third position of vector \( v_{npn} \)) denotes that current selected population \( P_{sel} \) includes four individuals with \( e_{npn}(3) = 1 \). This corresponds to the fact that the transistor NPN with C connected to \( n_1 \), B connected to \( n_2 \) and E connected to \( n_3 \) (see Tab. 3) was used four times in current selected population \( P_{sel} \). Similarly \( v_{res}(2) = 9 \) denotes that resistor connected to nodes 0 and 2 (see Tab. 1) was used nine times in \( P_{sel} \) and it becomes the most frequently used resistor in the individuals of current selected population \( P_{sel} \). In other words there is high probability that this resistor will appear in the topology of a good individuals in next generations.

After calculation of the marginal frequencies of the edges for all types of the components, the values of vectors \( v_{npn}, v_{pnp}, v_{res}, v_{vccp}, v_{vccn} \) are sorted from the highest to the lowest and this way vectors \( s_{npn}, s_{pnp}, s_{res}, s_{vccp}, s_{vccn} \) are obtained. Vectors of sorted marginal frequencies \( s_{npn}, s_{pnp}, s_{res}, s_{vccp}, s_{vccn} \) are used for determination of the most probable components during the phase of generation of new individuals (sampling phase). Sorted information about the marginal frequencies of the used components in current population \( P_{sel} \) stored in five vectors \( s_{npn}, s_{pnp}, s_{res}, s_{vccp}, s_{vccn} \) is denoted as probabilistic model \( M \).

6. Sampling of the Probabilistic Model

Created probabilistic model \( M \) is used to generate new solutions of the promising topologies of the given solution space. To increase diversity of the created samples some portion of the edges of the generated samples is added randomly. Presented sampling method was inspired by sampling principle of Estimation of Distribution Algorithm based on graph kernels presented in [3]. Pseudo-code of the used sampling method is presented in Fig. 12.

\[ \text{step1: Randomly select individual } I \text{ of population } P_{sel}. \]
\[ \text{step2: Randomly with probability } P_{rem} \text{ remove edges of graphs } G_{res}, G_{vccp}, G_{vccn} \text{ and hyperedges of hypergraphs } G_{npn}, G_{pnp} \text{ of individual } I. \]
\[ \text{step3: Add edges to graphs } G_{res}, G_{vccp}, G_{vccn} \text{ and hyperedges to hypergraphs } G_{npn}, G_{pnp} \text{ of selected individual } I. \]

Fig. 12. Flow chart of the sampling phase.

In step1 individual \( I \) of current selected population \( P_{sel} \) is chosen randomly and is used as a basis for the new generated sample.

In step2 the edges of graphs \( G_{res}, G_{vccp}, G_{vccn} \) and the hyperedges of hypergraphs \( G_{npn}, G_{pnp} \) of individual \( I \) are removed randomly with probability \( P_{rem} \) which is typically set to 0.2. In other words approximately 100. \( P_{rem} \) percent of the edges of graphs \( G_{res}, G_{vccp}, G_{vccn} \) and the hyperedges of hypergraphs \( G_{npn}, G_{pnp} \) of individual \( I \) are removed.

In step3 new edges are added to graphs \( G_{res}, G_{vccp}, G_{vccn} \) and new hyperedges are added to hypergraphs \( G_{npn}, G_{pnp} \). There are two ways how to perform this step. In the first one the process of the addition of the edges and the hyperedges is guided using information about the promising areas of the solution space stored in probabilistic model \( M \). The edges and the hyperedges with high values of the marginal frequencies in vectors \( s_{npn}, s_{pnp}, s_{res}, s_{vccp}, s_{vccn} \) are more favorable than those with lower values. This way modification of the topologies of graphs \( G_{res}, G_{vccp}, G_{vccn} \) and hypergraphs \( G_{npn}, G_{pnp} \) is guided to include the edges which are frequently used in the good individuals of the population. The second way is random addition of the edges and the hyperedges what helps to maintain diversity of the generated samples. Probability of using of probabilistic model \( M \) to guide the process of addition of the edges and the hyperedges is defined as \( P_{add} \) and is typically set to 0.8.

7. Objective Function

Information about the topology stored in the individuals of population \( P_{samp} \) and information about the parameters stored in parameters storage \( PS \) are transformed into netlist representation suitable for external spice compatible circuit simulator. The presented problem was synthesized using circuit simulator ngspice.

After obtaining of the voltage transfer characteristic cost value is calculated using objective function (2). To enable direct comparison of the results obtained using the proposed method to the results of other authors the objective function is defined exactly the same way as was presented in the original paper [1],

\[ \text{cost} = \sum_{i=1}^{m} w(i) | f_d(i) - f_c(i) |. \]  

According to (2) cost value \( \text{cost} \) is defined as weighted sum of absolute values of differences between voltage response of desired solution \( f_d \) and voltage response of current solution \( f_c \) over \( m = 21 \) equidistant voltage values in range -250 mV to 250 mV. There is penalization of the cost value by 10 if the output voltage response is not within 1% deviation of the target voltage characteristic. In such case weight \( w \) is set to 10, otherwise \( w = 1 \).

8. Parameters Optimization

In the last phase of the proposed method the parameters of resistors stored in parameters storage \( PS \) are optimized according to the topologies of \( n_{rpm} \) randomly selected individuals of newly created population \( P \). In every generation of the proposed method the parameters optimization is executed with probability \( P_{opt} \). Pseudo-code of the parameters optimization phase is presented in Fig. 13.
step1: Randomly choose individual $I$ of population $P$.
step2: Based on topology of individual $I$ load parameters $p_1$ from parameters storage $PS$.
step3: Using topology information stored in $I$ and parameters $p_1$ evaluate cost value of individual $I$.
step4: Execute local search algorithm. Optimized parameters $p_2$ and cost value of optimized solution $c_2$ are obtained.
step5: If $c_2 < c_1$ then replace parameters $p_1$ in $PS$ with parameters $p_2$.

Fig. 13. Pseudo code of the optimization phase.

Individual $I$ of current population $P$ is selected randomly (step1). Based on the topology encoded in individual $I$ corresponding parameters $p_1$ of parameters storage $PS$ are loaded and cost value $c_1$ of individual $I$ is evaluated (step2, step3).

In step4 the local search algorithm (LSA) tries to improve accuracy of individual $I$. Parameters $p_1$ loaded in step2 are used as a starting point for LSA. After finishing LSA new optimized parameters $p_2$ and cost value of the optimized individual $c_2$ are obtained.

If LSA was successful in improving of cost value of $I$ ($c_2 < c_1$) then parameters $p_1$ in parameters storage $PS$ are replaced by optimized parameters $p_2$. If LSA was not successful in improving accuracy of individual $I$ then the parameters optimization process is terminated with no modification of parameters storage $PS$ (step5).

As LSA Matlab function fmincon was used. The function was configured to use Interior-Point algorithm and maximal number of function evaluations $MaxFunEvals$ was set to 800. Parameters optimization method and its parameters were chosen based on two contradictory demands - low number of objective function evaluations of the whole algorithm and good accuracy of the solutions. Selected LSA method and its parameters ($MaxFunEvals$, ...) allow to achieve good compromise between both mentioned demands.

According to [2] the values of the resistors are chosen from E12 series in five decades. Thus resistance of every resistor can be set to one of 60 possible values. The lowest and the highest possible values of resistors were 10 $\Omega$ and 820 k$\Omega$ respectively.

9. Experiments and Solutions

The proposed algorithm was implemented in 64-bit version of Matlab 8.0 (R2012b). Experiments were performed on 64-bit dual core PC with processor AMD Athlon II X2 245, 8GB RAM and operational system Centos 6.5.

Number of total objective function evaluations $n_{evals}$ consists of number of objective function evaluations required by evaluation of cost values of $P_{amp}$ (step4 in Fig. 1) and number of objective function evaluations required by the parameters optimization phase (step6 in Fig. 1) and can be computed as $n_{evals} = n_{gen}d + n_{gen}P_{opt}MaxFunEvals$.

The parameters of the algorithm were set as follows. Maximal number of: nodes $n_{nod} = 17$, resistors $n_{res} = 12$, transistors NPN $n_{npn} = 14$, transistors PNP $n_{npn} = 14$, nodes connected to Vccp $n_{vccp} = 6$, nodes connected to Vccn $n_{vccn} = 6$. Size of population $P = m = 400$ individuals, size of population $P_{amp} d = 200$ individuals, generations per run $n_{gen} = 3000$, number of total objective function evaluations $n_{evals} = 1.5e6$, probability of execution of the parameters optimization $P_{opt} = 0.15$, number of optimized individuals $n_{opt} = 4$, number of objective function evaluations required by LSA $MaxFunEvals = 500$. These parameters were chosen experimentally. The goal was to achieve solutions of better accuracy with less number of required objective function evaluations than presented in [1] and [2].

The proposed algorithm was executed in four parallel threads. Five runs per single thread. Therefore 20 runs of the proposed algorithm in total. Average run time of a single run was 14 hours. Average time of a single evaluation of the objective function was 0.0336 second. Results of the runs are presented in Tab. 4.

![Table 4. Results of 20 runs of the proposed algorithm.](image)

The best solution was synthesized in run 3 of thread 3. Comparison of the output characteristics of the best solution and desired function (1) is presented in Fig. 14. Since both curves in Fig. 14 are almost merged together, deviation of $U_2$ is presented in Fig. 15. Netlist of the best solution obtained in the proposed experiments is presented in Fig. 17. Bipolar transistors NPN and PNP are denoted as bjtnpn and bjtpnp respectively. Default models were used for both types of the transistors. To reduce convergence problems caused by unconnected components and dangling terminals all nodes of the encoded analog circuit are connected to GND (node 0) through resistance 1 $\Omega$ (resistors Rg1 to Rg16). Resistors $R_{in}$ and $R_L$ are input and output resistances respectively and are set to 1 $k\Omega$. Schematic corresponding to the evolved netlist of the best solution in Fig. 17 is presented in Fig. 16. Since transistors q1 and q11 have no function in the synthesized circuit (netlist in Fig. 17) these transistors were not used in the resulting schematic (Fig. 16). Voltage $V_{IN}$ and voltage on resistor $R_L$ are input and output respectively.
Fig. 14. Comparison of output voltage characteristic $U_2 = f(U_1)$ of the best solution and desired function (1).

Fig. 15. Deviation of output voltage characteristic $U_2 = f(U_1)$ of the best solution and function (1).

Fig. 16. Schematic of the best solution (solution 3 in thread 3).
Fig. 17. Netlist of the best solution (solution 3 in thread 3).

10. Comparison to Other Methods

As was stated above the problem of circuit realization of cube root function which was introduced by Koza et al. in [1] was adopted also in [2]. Koza et al. [1] employed genetic programming (GP) approach. In [2] unconstrained genetic algorithm with oscillating length representation (GA OLG) was used. Comparison of the best solutions of both authors and the best solution of proposed method GhEDA is presented in Tab. 5.

<table>
<thead>
<tr>
<th>method</th>
<th>best cost</th>
<th>objective function evaluations</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP</td>
<td>1.68</td>
<td>37e6</td>
</tr>
<tr>
<td>GA OLG</td>
<td>2.27</td>
<td>4e6</td>
</tr>
<tr>
<td>GhEDA</td>
<td>1.44</td>
<td>1.5e6</td>
</tr>
</tbody>
</table>

Tab. 5. Comparison of the results of proposed method GhEDA to GP and GA OLG.

As can be seen in Tab. 5 proposed method GhEDA overperforms other two methods in terms of accuracy of the solution and number of required objective function evaluations as well.

Comparison of the number of the components of the best synthesized circuits of methods GP, GA OLG and GhEDA is presented in Tab. 6.

<table>
<thead>
<tr>
<th>method</th>
<th>GP</th>
<th>GA OLG</th>
<th>GhEDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of transistors</td>
<td>36</td>
<td>24</td>
<td>14</td>
</tr>
<tr>
<td>number of resistors</td>
<td>12</td>
<td>12</td>
<td>7</td>
</tr>
<tr>
<td>number of diodes</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

Tab. 6. Comparison of the number of the components of the best solutions of methods GP, GA OLG and GhEDA.

As can be seen from Tab. 6 the complexity of the synthesized circuit was highest for GP. Method GA OLG was able to reach circuit of lower complexity compared to GP. The best result was achieved using GhEDA method which was able to synthesize circuit twice smaller than circuit produced using GP.

11. Conclusion

There was presented graph based hybrid estimation of distribution algorithm (GhEDA) whose synthesis capability was demonstrated on the problem of circuit realization of cube root function. Results of the proposed method were compared to results of Koza et al. [1] (GP) and Sapargaliyev and Kalganova [2] (GA OLG) who adopted the same problem of synthesis of analog circuit realization of cube root function. Experiments have shown that in terms of accuracy of the solution and number of required objective function evaluations the proposed method overperforms both other methods.

The proposed method employs simple univariate probabilistic model based on the assumption that there are no dependencies between the variables of the solution vector. Although the presented experiments have shown that the used probabilistic model was suitable for the proposed method this model can be replaced by more advanced multivariate probabilistic model which is capable to capture higher order dependencies between the variables of the solution vector. This could be interesting and promising area of another research. Since some multivariate models can incorporate some portion of previous knowledge (prior) another interesting area of the research could be usage of different priors based on the target application of the synthesized circuit.

Since the proposed method is population based evolutionary algorithm, multiobjective approach as pareto ranking can be incorporated into the method. Also parallel computation of the cost values of the individuals of the population can be applied.

Acknowledgements

The research described in this paper received support through specific research project FEKT-S-14-2281. This work has also received funds from the operational program SIX denoted as CZ.1.05/2.1.00/03.0072. Publication of the results was financially supported by the project Popularization of BUT R&D Results and Support of Systematic Collaboration with Czech Students, no. CZ.1.07/2.3.00/35.0004.

References

[1] KOZA, J. R., BENNETT, F. H., FORREST, H., LOHN, J., DUNLAP, F., ANDRE, D., KEANE, M. A. Automated synthesis of computa-


About Authors ...

Josef SLEZÁK was born in Zlín, Czech Republic, in 1982. His research interest is circuit theory and evolutionary synthesis of analog circuits. He received the MSc. degree from the Brno University of Technology in 2007. Now he is working towards a PhD. degree at Department of Radio Electronics, Brno University of Technology.

Jiří PETRŽELA was born in Brno, Czech Republic, in 1978. He received the MSc. and PhD. degrees from the Brno University of Technology in 2003 and 2007 respectively. His research interest covers the nonlinear dynamics, chaos theory and analog circuit design. Currently he is an Associate Professor at the Department of Radio Electronics.